Home | Login | Logout | Access Informatio
Siter

Welcome United States Patent and Trademark Office

Search Results

BROWSE SEARCH IEEE XPLORE GUIDE

Results for **"(((fork* I branch* register allocat* parallel*)<in>metadata)) <and> (pyr >= 1980 <..."** e-mail
Your search matched **0** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» Search Options

View Session History

New Search

Modify Search

(((fork* I branch* register allocat* parallel*)<in>metadata)) <and> (pyr >=

□ Check to search only within this results set

Display Format: ● Citation ○ Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer to the Help pages assistance revising your search.

Help Contact U
Securi

Indexed by
Inspec®

Google Scholar BETA

fork conditional branch register allocation para Search

Advanced Scholar Search
Scholar Preferences
Scholar Help

Scholar Results **1 - 10** of about **277** for **fork conditional branch register allocation parallel code**. (0.(

Compiler-controlled multithreading for lenient **parallel** languages
KE Schauser, DE Culler, T Von Eicken - 1991 - cs.cornell.edu
... locate the next activation and **fork** to a ... theentry to a thread and **conditional** execution occurs ... expected quantum boundaries, frame and **register** assignment under ...
Cited by 43 - View as HTML - Web Search - portal.acm.org - portal.acm.org - all 4 versions » - Library Search

Static Analysis for Guarded **Code**
P Hu - LCR, 2000 - springerlink.com
... guarded **code** back to an explicit **conditional branch** structure where ... consists of predicate variables(ie **branch** conditions, eg p ... g 2 = dnf(l 1 ∧ l 2)} **(fork)** ...
Cited by 2 - Web Search - inria.fr - inria.fr - portal.acm.org

[PS] Static speculation, dynamic resolution
A Unger, T Ungerer, E Zehendner - Proc. 7th Workshop Compilers for **Parallel** Computers, 1998 - informatik.uni-augsburg.de
... compare instruction attached to the original **branch** remains in ... and moving them across the **fork** instruction ... 4. Selection of the **conditional** branches that cannot ...
Cited by 4 - View as HTML - Web Search

Utilising **Parallel** Resources by Speculation
A Unger, E Zehendner, T Ungerer - **Parallel** and Distributed Processing, 1999. PDP'99. ..., 1999 - ieeexplore.ieee.org
... Each **branch** is replaced by a **fork** instruc- tion ... instruction attached to the original **branch** remains in the ... Selection of the **conditional** branches that cannot be ...
Web Search - doi.ieeecs.org - doi.ieeecomputersociety.org - informatik.uni-augsburg.de - all 5 versions »

Program Structure a Basis for Parallelizing Global **Register**
A Zobel - ieeexplore.ieee.org
... The background and basic definitions for **register allocation** are presented ... to simplify non-interval **register** conflict graphs ... s; every loop or **conditional** has a ...
Web Search

Assigning confidence to **conditional branch** predictions
E Jacobsen, E Rotenberg, JE Smith - PROC ANNU INT SYMP MICROARCHITECTURE, 1996 - doi.ieeecs.org
... threads at any given time and to **fork** a second ... effect, executes the same number of **conditional** branches. ... of global **branch** outcomes in a **branch** history **register** ...
Cited by 130 - Web Search - tinker.ncsu.edu - american.cs.ucdavis.edu - ece.wisc.edu - all 14 versions »

Using Global **Code** Motions to Improve the Quality of Results for High-Level Synthesis
SGN Savoiu, NDRGA Nicolau, T Report - cecs.uci.edu

... of the operation into both the true and the false **branch** of a **conditional**. ... shown in Figure 3(d). The ability to duplicate operations across **fork** (or **branch** ...
View as HTML - Web Search - ics.uci.edu - mesl.ucsd.edu - ics.uci.edu

Evaluation of Mechanisms for Fine-Grained **Parallel** Programs in the J-Machine and the CM-5
E Spertus, SC Goldstein, KE Schauser, T von Eicken ... - ACM SIGARCH Computer Architecture News, 1993 - portal.acm.org
... as **register allocation**, and there is no external scheduler ... with in- lets using a new **register** window ... introduce annulling branches and **branch** delay slots into the ...
Cited by 28 - Web Search - newit.gsu.unibel.by - cs.cornell.edu - ieeexplore.ieee.org - all 5 versions »

[PS] Global **Code** Selection of Directed Acyclic Graphs
A Fauth, G Hommel, A Knoll, C Mueller - CC, 1994 - wwwknoll.informatik.tu-muenchen.de
... modeling in-place storage of signals and the programming of the **branch** ... have a **conditional** context . ... We assume that the data- paths do not **fork** (and thus do not ...
Cited by 11 - View as HTML - Web Search - in.tum.de - atknoll1.informatik.tu-muenchen.de - portal.acm.org - all 7 versions »

CARS: A New **Code** Generation Framework for Clustered ILP Processors
K Kailas, K Ebcioglu, AK Agrawala - HPCA, 2001 - doi.ieeecomputersociety.org
... a) Source **code** φ ... a F **Fork** node DEF-USE edges J Join node prfrd_reg_map propagation path ... DEF-USE chains [40] – an ob- ject for **register allocation** in graph ...
Cited by 36 - Web Search - doi.ieeecs.org - e-kailas.net - portal.acm.org - all 8 versions »

Gooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

fork conditional branch register alloc Search

Google Home - About Google - About Google Scholar

PORTAL

USPTO

Subscribe (Full Service) Register (Limited Service, Free) Lo

Search: ● The ACM Digital Library ○ The Guide

+conditional +branch +fork +register +allocat*

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction su

Terms used **conditional branch fork register allocat**

Found **189** of **160**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 189

Result page: **1** 2 3 4 5 6 7 8 9 10 next

Relevance scale □ ▄ ▆

1 Compiler transformations for high-performance computing

David F. Bacon, Susan L. Graham, Oliver J. Sharp

December 1994 **ACM Computing Surveys (CSUR)**, Volume 26 Issue 4

Full text available: pdf(6.32 MB) Additional Information: full citation, abstract, references, citings, index terms, review

In the last three decades a large number of compiler transformations for optimizing programs have been implemented. Most optimizations for uniprocessors reduce the number of instructions executed by the program using transformations based on the analysis of scalar quantities and data-flow techniques. In contrast, optimizations for high-performance superscalar, vector, and parallel processors maximize parallelism and memory locality with transformations that rely on tracking the properties o ...

Keywords: compilation, dependence analysis, locality, multiprocessors, optimization, parallelism, superscalar processors, vectorization

2 A combined compiler and architecture technique to control multithreaded execution of branches and loop iterations

A. Unger, E. Zehendner, Th. Ungerer

March 2000 **ACM SIGARCH Computer Architecture News**, Volume 28 Issue 1

Full text available: pdf(930.42 KB) Additional Information: full citation, abstract, index terms

Simultaneous Speculation Scheduling (S^3) is a combined compiler and architecture technique to control multiple path execution. It can be used for dual path branch speculation in case of unpredictable branches and for multiple path speculative execution of loop iterations in case of loop-carried dependences that make parallel execution otherwise impossible. We apply S^3 In situations where purely static techniques cannot prove data independence. S^3

Keywords: dual path execution, eager execution, instruction scheduling, multithreading, speculation

3 A Survey of Some Theoretical Aspects of Multiprocessing

J. L. Baer
January 1973 **ACM Computing Surveys (CSUR),** Volume 5 Issue 1
Full text available: pdf(4.05 MB) Additional Information: full citation, references, citings, index terms

4 Bridge: a versatile behavioral synthesis system
Chia-Jeng Tseng, Ruey-Sing Wei, Steven G. Rothweiler, Michael M. Tong, Ajoy K. Bose
June 1988 **Proceedings of the 25th ACM/IEEE conference on Design automation**
Full text available: pdf(748.97 KB) Additional Information: full citation, abstract, references, citings, index terms

Bridge is a behavioral synthesis system being developed at AT&T Bell Laboratories. Two slicing techniques are implemented in this system to drive structural allocation; one is local slicing and the other is global slicing. Global slicing supports the synthesis of concurrent processes with a centralized control. A variable in a behavioral description can be either a storage element or a signal. The impacts of treating a variable as a signal on data flow scheduling, control flow schedulin ...

5 How datapath allocation affects controller delay
Steve C.-Y. Huang, Wayne H. Wolf
May 1994 **Proceedings of the 7th international symposium on High-level synthesis**
Full text available: pdf(559.88 KB) Additional Information: full citation, references, citings

6 Threaded multiple path execution
Steven Wallace, Brad Calder, Dean M. Tullsen
April 1998 **ACM SIGARCH Computer Architecture News , Proceedings of the 25th annual international symposium on Computer architecture,** Volume 26 Issue 3
Full text available: pdf(1.49 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

This paper presents *Threaded Multi-Path Execution* (TME), which exploits existing hardware on a Simultaneous Multi-threading (SMT) processor to speculatively execute multiple paths of execution. When there are fewer threads in an SMT processor than hardware contexts, threaded multi-path execution uses spare contexts to fetch and execute code along the less likely path of hard-to-predict branches.This paper describes the hardware mechanisms needed to enable an SMT processor to efficiently s ...

7 Execution-based prediction using speculative slices
Craig Zilles, Gurindar Sohi
May 2001 **ACM SIGARCH Computer Architecture News , Proceedings of the 28th annual international symposium on Computer architecture,** Volume 29 Issue 2
Full text available: pdf(1.03 MB) Additional Information: full citation, abstract, references, citings, index terms

A relatively small set of static instructions has significant leverage on program execution performance. These problem instructions contribute a disproportionate number of cache misses *and branch mispredictions because their behavior cannot be accurately anticipated using existing prefetching or branch prediction mechanisms.*

The behavior of many problem instructions can be predicted by executing a small code fragment called a speculative slice. If a speculative slice is exec ...

8 Parallel execution of prolog programs: a survey
Gopal Gupta, Enrico Pontelli, Khayri A.M. Ali, Mats Carlsson, Manuel V. Hermenegildo
July 2001 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 23 Issue 4
Full text available: ▨ pdf(1.95 MB) Additional Information: full citation, abstract, references, citings, index terms

Since the early days of logic programming, researchers in the field realized the potential for exploitation of parallelism present in the execution of logic programs. Their high-level nature, the presence of nondeterminism, and their referential transparency, among other characteristics, make logic programs interesting candidates for obtaining speedups through parallel execution. At the same time, the fact that the typical applications of logic programming frequently involve irregular computatio ...

Keywords: Automatic parallelization, constraint programming, logic programming, parallelism, prolog

9 Assigning confidence to conditional branch predictions
Erik Jacobsen, Eric Rotenberg, J. E. Smith
December 1996 **Proceedings of the 29th annual ACM/IEEE international symposium on Microarchitecture**
Full text available: ▨ pdf(1.28 MB) Additional Information: full citation, abstract, references, citings, index terms

Many high performance processors predict conditional branches and consume processor resources based on the prediction. In some situations, resource allocation can be better optimized if a confidence level is assigned to a branch prediction; i.e. if the quantity of resources allocated is a function of the confidence level. To support such optimizations, we consider hardware mechanisms that partition conditional branch predictions into two sets: those which are accurate a relatively high percentag ...

Keywords: branch correctness, conditional branch predictions, dynamic branches, processor resources, resource allocation, static branches

10 Multipath execution: opportunities and limits
Pritpal S. Ahuja, Kevin Skadron, Margaret Martonosi, Douglas W. Clark
July 1998 **Proceedings of the 12th international conference on Supercomputing**
Full text available: ▨ pdf(1.23 MB) Additional Information: full citation, references, citings, index terms

11 Assembly instruction level reverse execution for debugging

Tankut Akgul, Vincent J. Mooney III
April 2004 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 13 Issue 2
Full text available: pdf(1.18 MB) Additional Information: full citation, abstract, references, index terms

Assembly instruction level reverse execution provides a programmer with the ability to return a program to a previous state in its execution history via execution of a "reverse program." The ability to execute a program in reverse is advantageous for shortening software development time. Conventional techniques for recovering a state rely on saving the state into a record before the state is destroyed. However, state-saving causes significant memory and time overheads during forward execution.Th ...

Keywords: Debugging, reverse code generation, reverse execution

12 Fine-grain parallelism with minimal hardware support: a compiler-controlled threaded abstract machine
David E. Culler, Anurag Sah, Klaus E. Schauser, Thorsten von Eicken, John Wawrzynek
April 1991 **Proceedings of the fourth international conference on Architectural support for programming languages and operating systems**, Volume 19 , 25 , 26 Issue 2 , Special Issue , 4
Full text available: pdf(1.41 MB) Additional Information: full citation, references, citings, index terms

13 Register integration: a simple and efficient implementation of squash reuse
Amir Roth, Gurindar S. Sohi
December 2000 **Proceedings of the 33rd annual ACM/IEEE international symposium on Microarchitecture**
Full text available: pdf(154.98 KB) ps (573.81 KB) Publisher Site Additional Information: full citation, references, citings, index terms

14 ORBIT: an optimizing compiler for scheme
David Kranz, Richard Kelsey, Jonathan Rees, Paul Hudak, James Philbin
July 1986 **ACM SIGPLAN Notices , Proceedings of the 1986 SIGPLAN symposium on Compiler contruction**, Volume 21 Issue 7
Full text available: pdf(1.38 MB) Additional Information: full citation, references, citings, index terms

15 Fast breakpoints: design and implementation
Peter B. Kessler
June 1990 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1990 conference on Programming language design and implementation**, Volume 25 Issue 6

Full text available: pdf(855.02 KB) Additional Information: full citation, abstract, references, citings, index terms

We have designed and implemented a fast breakpoint facility. Breakpoints are usually thought of as a feature of an interactive debugger, in which case the breakpoints need not be particularly fast. In our environment breakpoints are often used for non-interactive information gathering; for example, procedure call count and statement execution count profiling [Swinehart, et al.]. When used non-interactively, breakpoints should be as fast as possible, so as to perturb the execution of the pro ...

16 Parameter passing and control stack management in Prolog implementation revisited
Neng-Fa Zhou
November 1996 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 18 Issue 6

Full text available: pdf(280.75 KB) Additional Information: full citation, abstract, references, citings, index terms

Parameter passing and control stack management are two of the crucial issues in Prolog implementation. In the Warren Abstract Machine (WAM), the most widely used abstract machine for Prolog implementation, arguments are passed through argument registers, and the information associated with procedure calls is stored in possibly two frames. Although accessing registers is faster than accessing memory, this scheme requires the argument registers to be saved and restored for back tracking and m ...

Keywords: abstract machine, prolog

17 Fortran 8X draft
Loren P. Meissner
December 1989 **ACM SIGPLAN Fortran Forum**, Volume 8 Issue 4

Full text available: pdf(21.36 MB) Additional Information: full citation, abstract, index terms

Standard Programming Language Fortran. This standard specifies the form and establishes the interpretation of programs expressed in the Fortran language. It consists of the specification of the language Fortran. No subsets are specified in this standard. The previous standard, commonly known as "FORTRAN 77", is entirely contained within this standard, known as "Fortran 8x". Therefore, any standard-conforming FORTRAN 77 program is standard conforming under this standard. New features can b ...

18 A new framework for debugging globally optimized code
Le-Chun Wu, Rajiv Mirani, Harish Patil, Bruce Olsen, Wen-mei W. Hwu
May 1999 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1999 conference on Programming language design and implementation**, Volume 34 Issue 5

Full text available: pdf(1.54 MB) Additional Information: full citation, abstract, references, citings, index terms

With an increasing number of executable binaries generated by optimizing compilers today, providing a clear and correct source-level debugger for programmers to debug optimized code has become a necessity. In this paper, a new framework for debugging globally optimized code is proposed. This framework consists of a new code location mapping scheme, a data location tracking scheme, and an emulation-based forward recovery model. By taking over the control early and emulating instructions selective ...

19 Architecture 2: Dual path instruction processing
Juan L. Aragón, José González, Antonio González, James E. Smith
June 2002 **Proceedings of the 16th international conference on Supercomputing**
Full text available: pdf(332.19 KB) Additional Information: full citation, abstract, references, index terms

The reasons for performance losses due to conditional branch mispredictions are first studied. Branch misprediction penalties are broken into three categories: pipeline-fill penalty, window-fill penalty, and serialization penalty. The first and third of these produce most of the performance loss, but the second is also significant. Previously proposed dual (or multi) path execution methods attempt to reduce all three penalties, but these methods are also quite complex. Most of the complexity is ...

Keywords: branch misprediction penalty, confidence estimation, dual path processing, pre-scheduling

20 Global scheduling independent of control dependencies based on condition vectors
K. Wakabayashi, H. Tanaka
July 1992 **Proceedings of the 29th ACM/IEEE conference on Design automation**
Full text available: pdf(477.57 KB) Additional Information: full citation, references, citings, index terms

Results 1 - 20 of 189 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Playe

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	2	(conver$4 or transform$3 or translat$3or compil$5) same ("conditional branch") near5 ((parallel$7 or multiple or multi) near5 thread)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/29 07:39
L2	2	("conditional branch") near5 ((parallel$7 or multiple or multi) near5 thread)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/29 07:12
L3	171	("conditional branch") near5 (fork$3 or spawn$3 or parallel$7 or multiple or multi)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/29 07:13
L4	16	(conver$4 or transform$3 or translat$3or compil$5) same ("conditional branch") near5 (fork$3 or spawn$3 or parallel$7 or multiple or multi)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/29 07:14
L5	50	("4819234" | "4872167" | "5168554" | "5301325" | "5333280" | "5450575" | "5504932" | "5533192" | "5557761" | "5564051" | "5581764" | "5594864" | "5598560" | "5632032" | "5652889" | "5712996" | "5742803" | "5754855" | "5768591" | "5768592" | "5774721" | "5787245" | "5805892" | "5812811" | "5826265" | "5867643" | "5877766" | "5887166" | "5901315" | "5903730" | "5913925" | "5953530" | "5961639" | "5966539" | "5978902" | "6002872" | "6002879" | "6009269" | "6029005" | "6049671" | "6058493" | "6059840" | "6072952" | "6094716" | "6101524" | "6112293" | "6128773" | "6151701" | "6151704").PN. OR ("6430676").URPN.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/29 07:23

L6	56	("4819234" | "4872167" | "5168554" | "5301325" | "5333280" | "5450575" | "5504932" | "5533192" | "5557761" | "5564051" | "5581764" | "5594864" | "5598560" | "5632032" | "5652889" | "5712996" | "5742803" | "5754855" | "5768591" | "5768592" | "5774721" | "5787245" | "5805892" | "5812811" | "5826265" | "5867643" | "5877766" | "5887166" | "5901315" | "5903730" | "5913925" | "5953530" | "5961639" | "5966539" | "5978902" | "6002872" | "6002879" | "6009269" | "6029005" | "6049671" | "6058493" | "6059840" | ("6072952" | "6094716" | "6101524" | "6112293" | "6128773" | "6151701" | "6151704").PN. OR ("6430676"). URPN.) and (register adj allocat$3)	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/29 07:26
L7	7	(("4819234" | "4872167" | "5168554" | "5301325" | "5333280" | "5450575" | "5504932" | "5533192" | "5557761" | "5564051" | "5581764" | "5594864" | "5598560" | "5632032" | "5652889" | "5712996" | "5742803" | "5754855" | "5768591" | "5768592" | "5774721" | "5787245" | "5805892" | "5812811" | "5826265" | "5867643" | "5877766" | "5887166" | "5901315" | "5903730" | "5913925" | "5953530" | "5961639" | "5966539" | "5978902" | "6002872" | "6002879" | "6009269" | "6029005" | "6049671" | "6058493" | "6059840" | "6072952" | "6094716" | "6101524" | "6112293" | "6128773" | "6151701" | "6151704").PN. OR ("6430676"). URPN.) and (register adj allocat$3)	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/29 07:27

<table>
<tr><th></th><th></th><th></th><th></th><th></th><th></th><th></th></tr>
<tr><td>L8</td><td>28</td><td>("5233696" | "5345569" | "5574935" | "5623628" | "5632023" | "5664215" | "5696955" | "5768610" | "5857089" | "5892936" | "5933618" | "5987592").PN. OR ("6094716").URPN.</td><td>US-PGPUB; USPAT; USOCR</td><td>OR</td><td>OFF</td><td>2005/08/29 07:33</td></tr>
<tr><td>L9</td><td>11</td><td>(("5233696" | "5345569" | "5574935" | "5623628" | "5632023" | "5664215" | "5696955" | "5768610" | "5857089" | "5892936" | "5933618" | "5987592").PN. OR ("6094716").URPN.) and (register adj allocat$3)</td><td>US-PGPUB; USPAT; USOCR</td><td>OR</td><td>OFF</td><td>2005/08/29 07:36</td></tr>
<tr><td>L10</td><td>0</td><td>coloring same register same ((conditional or predicate or boolean) adj branch)</td><td>US-PGPUB; USPAT; USOCR</td><td>OR</td><td>OFF</td><td>2005/08/29 07:38</td></tr>
<tr><td>L11</td><td>1</td><td>coloring same ((conditional or predicate or boolean) adj branch)</td><td>US-PGPUB; USPAT; USOCR</td><td>OR</td><td>OFF</td><td>2005/08/29 07:39</td></tr>
<tr><td>L12</td><td>127</td><td>(creat$3 or generat$3) near5 parallel$7 and ((register and allocat$3) or "graph coloring" or live$6) same branch$3</td><td>US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB</td><td>OR</td><td>OFF</td><td>2005/08/29 07:41</td></tr>
<tr><td>L13</td><td>3</td><td>(creat$3 or generat$3) near5 parallel$7 same ((register and allocat$3) or "graph coloring" or live$6) same branch$3</td><td>US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB</td><td>OR</td><td>OFF</td><td>2005/08/29 07:41</td></tr>
<tr><td>L14</td><td>16</td><td>(creat$3 or generat$3) near5 parallel$7 same ((register and allocat$3) or "graph coloring" or live$6) same (branch$3 or loop$3)</td><td>US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB</td><td>OR</td><td>OFF</td><td>2005/08/29 08:13</td></tr>
<tr><td>L15</td><td>250</td><td>717/159.ccls.</td><td>US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB</td><td>OR</td><td>OFF</td><td>2005/08/29 08:14</td></tr>
<tr><td>L16</td><td>0</td><td>717/159.ccls. and (register adj allocat) same (branch$3 or loop$3)</td><td>US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB</td><td>OR</td><td>OFF</td><td>2005/08/29 08:14</td></tr>
<tr><td>L17</td><td>27</td><td>717/159.ccls. and (register adj allocat$3) same (branch$3 or loop$3)</td><td>US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB</td><td>OR</td><td>OFF</td><td>2005/08/29 08:14</td></tr>
</table>

L18	5	717/149.ccls. and (register adj allocat$3) same (branch$3 or loop$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/29 08:15
L19	178	717/149.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/29 08:15
S1	0	"6622301".pn. and register	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/05/12 12:02
S2	0	"6622301".pn. and register	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 12:02
S3	1	"6622301".pn. and instruction	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 12:10
S4	0	"6622301".pn. and (basic adj block)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 12:10
S5	2	"6622301".pn. and (block)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 13:09
S6	0	"6622301".pn. and (branch$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 12:11
S7	0	"6622301".pn. and (distance near5 dependence)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 13:09
S8	0	"6622301".pn. and (distance)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 13:10

S9	1	"6622301".pn. and (locality)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 14:33
S10	999	(control adj flow) same (data adj flow) and (conver$4 or switch$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 14:34
S11	668	(control adj flow) same (data adj flow) and (conver$4 or switch$3) and parallel$7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 14:35
S12	228	(control adj flow) same (data adj flow) and (conver$4 or switch$3) same parallel$7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 15:08
S13	9	(control adj flow) same (data adj flow) and (conver$4 or switch$3) same parallel$7 and speculat$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 14:45
S14	21	(control adj flow) same (data adj flow) same (conver$4 or switch$3) same parallel$7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 15:08
S15	0	(control adj flow) same (data adj flow) same (conver$4 or switch$3) same parallel$7 and (register adj allocation)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 15:08
S16	203	(control adj flow) same (data adj flow) and (conver$4 or switch$3) same parallel$7 and (register ald allocation)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 15:09
S17	9	(control adj flow) same (data adj flow) and (conver$4 or switch$3) same parallel$7 and (register adj allocation)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 15:10
S18	9	(control adj flow) same (data adj flow) and (conver$4 or switch$3) same parallel$7 and (register adj allocation)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 15:13

S19	0	"5448737".pn. and register	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 15:29
S20	1	"6588009".pn. and register	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 15:31
S21	2	"6588009".pn. and resource	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 16:33
S22	10	("5598561" "6588009" "6292939" "6622301" "6725448").PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/05/12 16:33
S23	45	(conver$4 or transform$3 or translat$3) same branch same ((parallel$7 or multiple or multi) near5 thread)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/29 07:10
S24	24	(optimiz$5 or optimis$5) near5 branch same (parallel$7 or multi-thread or multi-processor or (plurallity near3 processor))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/20 15:09
S25	31	(conver$4 or transform$3 or translat$3) same branch same ((parallel$7 or multiple or multi) near5 thread) and (control or data)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/16 15:33
S26	20	(optimiz$5 or optimis$5) near5 branch same (parallel$7 or multi-thread or multi-processor or (plurallity near3 processor)) and (control or data)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/20 15:15
S27	7	(optimiz$5 or optimis$5) near5 branch same (parallel$7 or multi-thread or multi-processor or (plurallity near3 processor)) and (control or data) and (profil$3 or instrument$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/20 15:10
S28	184	generat$3 near5 (parallel$7 adj code)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/20 15:16

S29	5	generat$3 near5 (parallel$7 adj code) same target	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/20 15:16
S30	35	generat$3 near5 (parallel$7 adj code) and (optimiz$7 or optimis$7 or optimal)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/20 15:17
S31	37	generat$3 near5 (parallel$7 adj (code or instructions)) and (optimiz$7 or optimis$7 or optimal)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/20 15:18
S32	66	(generat$3 or creat$3 or produc$3 or output$4) near5 (parallel$7 adj (code or instructions)) and (optimiz$7 or optimis$7 or optimal)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/20 16:19
S33	1	"1107116".URPN.	USPAT	OR	OFF	2004/04/20 15:27
S34	8	("5237691" | "5317743" | "5347639" | "5408658" | "5412784" | "5515535" | "5732234" | "5857180").PN.	USPAT	OR	OFF	2004/04/20 15:46
S35	0	"6243863".URPN.	USPAT	OR	OFF	2004/04/20 15:50
S36	3	"6339840".URPN.	USPAT	OR	OFF	2004/04/20 15:52
S37	214	717/146.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/20 16:30
S38	225	717/151.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/20 16:30
S39	201	717/158.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/20 16:30
S40	71	717/161.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/20 16:30

S41	152	717/149.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/22 08:36
S42	555	muscat	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/22 08:43
S43	35	muscat and branch$3 and parallel$7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/22 08:43
S44	2	"6330661".URPN.	USPAT	OR	OFF	2004/04/22 08:40
S45	0	"6687812".URPN.	USPAT	OR	OFF	2004/04/22 08:41
S46	35	muscat and branch$3 and parallel$7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/22 08:43
S47	9	muscat and 7??/???.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/22 08:44
S48	429	parallel$7 and (register adj allocation)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/22 08:44
S49	195	parallel$7 and (register adj allocation) and branch$3 and (metric or probabilities or probability or statistics or measurements or analysis)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/22 08:46
S50	5	parallel$7 same (register adj allocation) same branch$3 and (metric or probabilities or probability or statistics or measurements or analysis)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/22 11:16
S51	1185	popescu.in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/22 11:16

S52	0	popescu.in. and register adj allocat$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/22 11:16
S53	15	popescu.in. and register	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/22 11:17
S54	9	popescu.in. and register and parallel$7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/22 11:19
S55	13	(register adj allocat$3) near5 parallel$7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/22 11:25
S56	0	(register adj allocat$3) near5 simd	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/22 11:27
S57	0	(register adj allocat$3) near5 (multi-processor)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/22 11:27
S58	56	register near5 (multi-processor)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/22 14:18
S59	1	register near5 (multi-processor) and code adj generat$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/22 11:30
S60	2	"6330661".URPN.	USPAT	OR	OFF	2004/04/22 11:43
S61	3	("5913059" | "5996068" | "6092175").PN.	USPAT	OR	OFF	2004/04/22 11:44
S62	955	parallel$7 near3 compil$5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/04/22 14:18

S63	110	parallelizing adj compiler	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/04/22 14:19
S64	19	parallelizing adj compiler same (optimiz$5 or optimis$5 or optimal$2)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/04/22 14:19
S65	4	("6622301" "6588009").pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/16 15:29
S66	433	(register adj allocation) and (trial or estimat$5 or heuristic or analysis or analyz$3) and (intermediate or il or graph or tree or platform-independent or abstract)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/11/16 15:33
S67	46	(conver$4 or transform$3 or translat$3 optimiz$5 or optimis$5) same (branch or conditional or boolean) same ((parallel$7 or multiple or multi) near5 thread) and (control or data)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/16 15:40
S68	1	S66 and S67.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/16 15:35
S69	61	(conver$4 or transform$3 or translat$3 optimiz$5 or optimis$5) same (branch or conditional or boolean) same ((parallel$7 or multiple or multi) near5 thread)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/16 15:41
S70	61	(conver$4 or transform$3 or translat$3 or optimiz$5 or optimis$5) same (branch or conditional or boolean) same ((parallel$7 or multiple or multi) near5 thread)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/16 15:42
S71	1	S66 and S70	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/16 15:41

S72	5555	(conver$4 or transform$3 or translat$3 or optimiz$5 or optimis$5) same (branch or conditional or boolean) same (parallel$7 or multiple or multi)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/16 15:42
S73	43	S66 and S72	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/17 08:29
S74	12	(register near2 heuristics) and parallel$7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/16 15:49
S75	0	("6321379").URPN.	USPAT	OR	OFF	2004/11/16 16:01
S76	0	("6430676").URPN.	USPAT	OR	OFF	2004/11/16 16:01
S77	1	("6430676").PN.	USPAT	OR	OFF	2004/11/16 16:01
S78	0	("6430676").URPN.	USPAT	OR	OFF	2004/11/16 16:01
S79	49	("4819234" | "4872167" | "5168554" | "5301325" | "5333280" | "5450575" | "5504932" | "5533192" | "5557761" | "5564051" | "5581764" | "5594864" | "5598560" | "5632032" | "5652889" | "5712996" | "5742803" | "5754855" | "5768591" | "5768592" | "5774721" | "5787245" | "5805892" | "5812811" | "5826265" | "5867643" | "5877766" | "5887166" | "5901315" | "5903730" | "5913925" | "5953530" | "5961639" | "5966539" | "5978902" | "6002872" | "6002879" | "6009269" | "6029005" | "6049671" | "6058493" | "6059840" | "6072952" | "6094716" | "6101524" | "6112293" | "6128773" | "6151701" | "6151704").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2004/11/16 16:02
S80	9041	itou.in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/17 08:30

S81	3	itou.in. and (barrier adj instruction)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/17 08:30
S82	0	("6292939").URPN.	USPAT	OR	OFF	2004/11/17 08:38
S83	10	("5535393" | "5778423" | "5781775" | "5802374" | "5873105" | "5953736" | "6016505" | "6081665" | "6098089" | "6216174").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2004/11/17 09:18
S84	1228	torii.in.	US-PGPUB; USPAT; USOCR	OR	OFF	2004/11/17 09:19
S85	450	torii.in. and parallel	US-PGPUB; USPAT; USOCR	OR	OFF	2004/11/17 09:19
S86	24	torii.in. and (parallel near5 control)	US-PGPUB; USPAT; USOCR	OR	OFF	2004/11/18 07:14
S87	2	("6389446").URPN.	USPAT	OR	OFF	2004/11/17 09:26
S88	1	("6622155").URPN.	USPAT	OR	OFF	2004/11/17 09:28
S89	80	sakai.in. and (parallel near5 control)	US-PGPUB; USPAT; USOCR	OR	OFF	2004/11/17 09:34
S90	470	muscat	US-PGPUB; USPAT; USOCR	OR	OFF	2004/11/18 07:14
S91	153	muscat and parallel$7	US-PGPUB; USPAT; USOCR	OR	OFF	2004/11/18 07:15
S92	0	("6687812").URPN.	USPAT	OR	OFF	2004/11/18 07:18
S93	8	("5717926" | "5724565" | "5812811" | "5958047" | "5961639" | "6065115" | "6304960" | "6330662").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2004/11/18 07:18
S94	247463	register near "3" (trial or profil$3) and parallel$7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/22 10:06
S95	66346	register near "3" (trial or profil$3) same parallel$7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/22 10:06

S96	26	register near3 (trial or profil$3) same parallel$7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/22 10:08
S97	16	register near3 (trial or profil$3) and parallel$7 and 717/???	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/22 10:26
S98	2	(register adj allocat$3) near5 (trial or profil$3) and (parallel$7 or branch$3 or fork$3) and 717/???	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/22 10:47
S99	1	(register adj allocat$3) near5 (trial or profil$3) and (parallel$7 or branch$3 or fork$3) and dependen$2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/22 10:28
S100	2	(register adj allocat$3) near5 (trial or profil$3) and (parallel$7 or branch$3 or fork$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/22 11:12
S101	96	(register adj allocat$3) near5 (trial or profil$3 or optimi$6) and (parallel$7 or branch$3 or fork$3 or (control near3 speculat$3))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/22 11:13
S102	2	(register adj allocat$3) near5 (trial or profil$3) and (parallel$7 or branch$3 or fork$3 or (control near3 speculat$3))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/11/22 11:13